FACULTE DES SCIENCES ET
TECHNIQUES DE L'INGENIEUR

AUTONOMOUS SYSTEM LAB 2 (ASL2)

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

User Guide

FPGA Turret For Khepera Mobile Robot

Stéphane Hofmann

February 2003

Content

1. FPGA Module Architecture

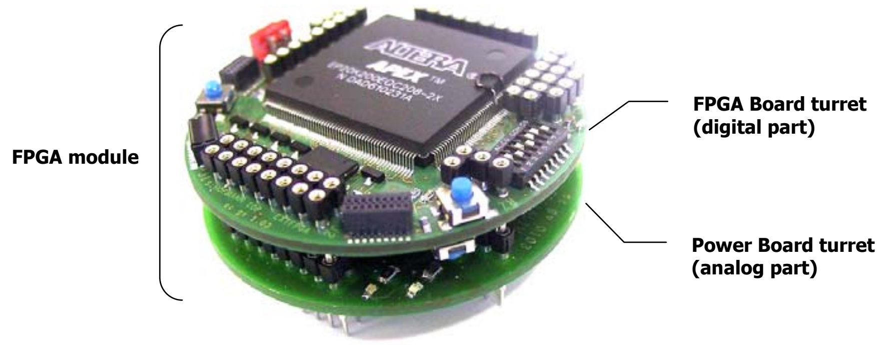

FPGA module
FPGA Board turret (digital part)
Power Board turret (analog part)

Figure 1: FPGA module picture

The system is composed of two different functional parts:

1. The **digital part (FPGA Board)**, which includes the major components as the FPGA, the CPLD, the memories, the oscillator, the transceiver, …

2. The **analog part (Power Board)**, which comprises the voltage regulators that generates the 1.8V and the 3.3V needed to supply the digital part.

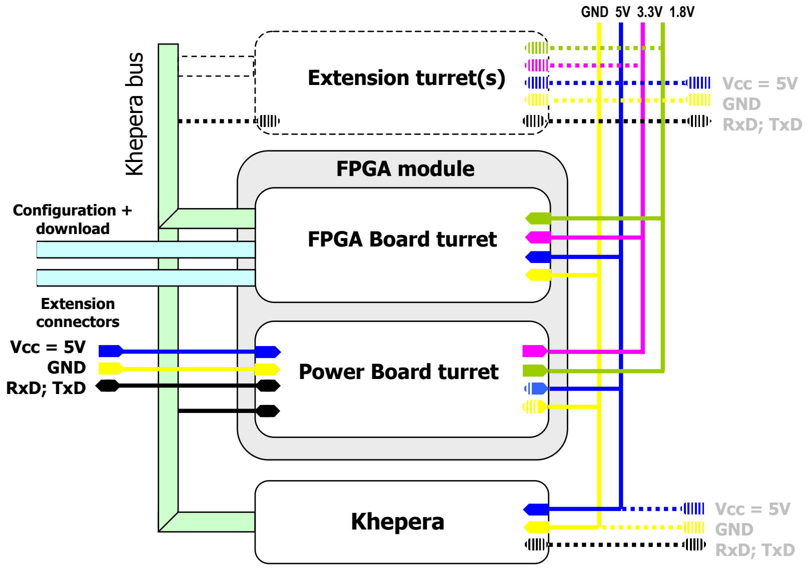


Figure 2 : Global architecture of FPGA module

2. FPGA Board Turret

2.1 Architecture

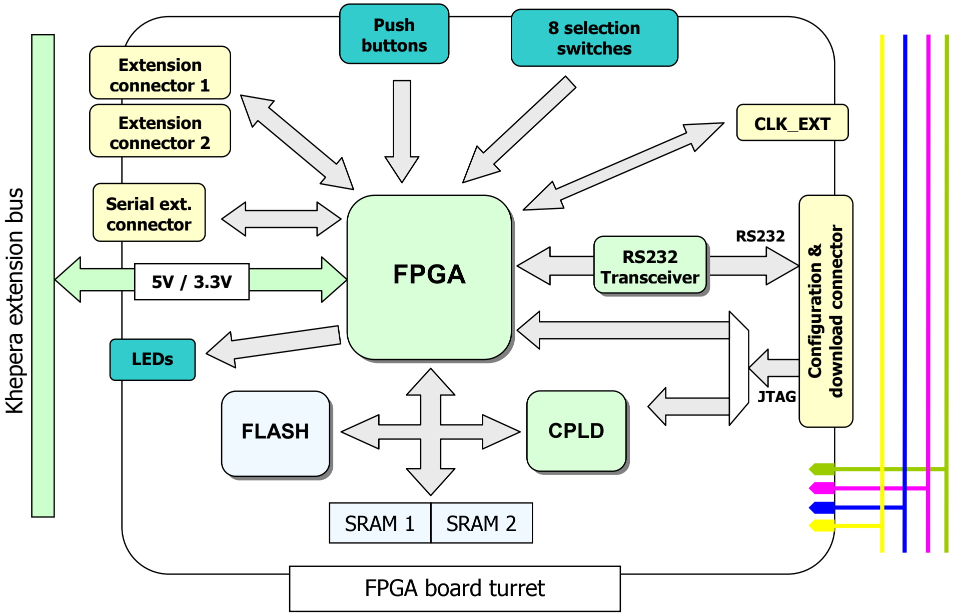


Figure 3 : FPGA Board Architecture

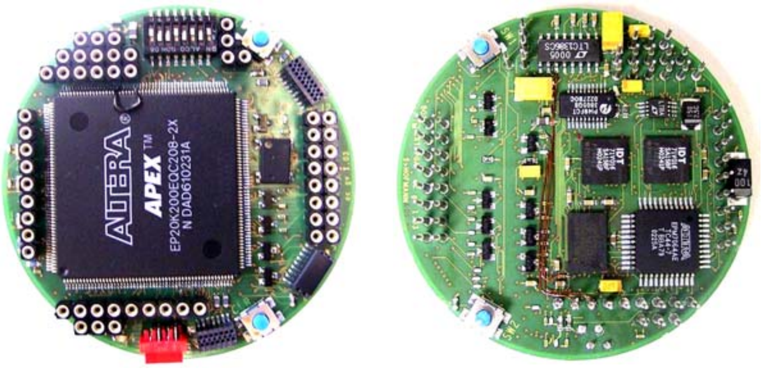


Figure 4 : Pictures of FPGA Board turret (left : top side, right : bottom side)

2.2 Features

- An APEX 20K200E(-2X) device (FPGA)
- 1 Mbyte (512 K x 16-bit) of Flash memory
  - pre-configured with a 32-bit Nios reference design and software
- 256 Kbytes of SRAM (in 2 64K x 16-bit chips)
- On-board logic for configuring APEX device from flash memory
- 3.3V compact extension connector (access to 22 user I/O)
- 5V-tolerant extension MicroMaTch connector (Serial connection or 2 user I/O)
- One RS-232 serial port
- Two user-definable push-button switches
- Three user-definable LEDs
- Joint Test Action Group (JTAG) connector for ByteBlasterMV and MasterBlasters Programmers
- Oscillator and zero skew clock distribution circuitry
- Power-on reset circuitry
- Excalibur development board compatible peripherals
- Excalibur development board compatible software development Kit

2.3 Layout

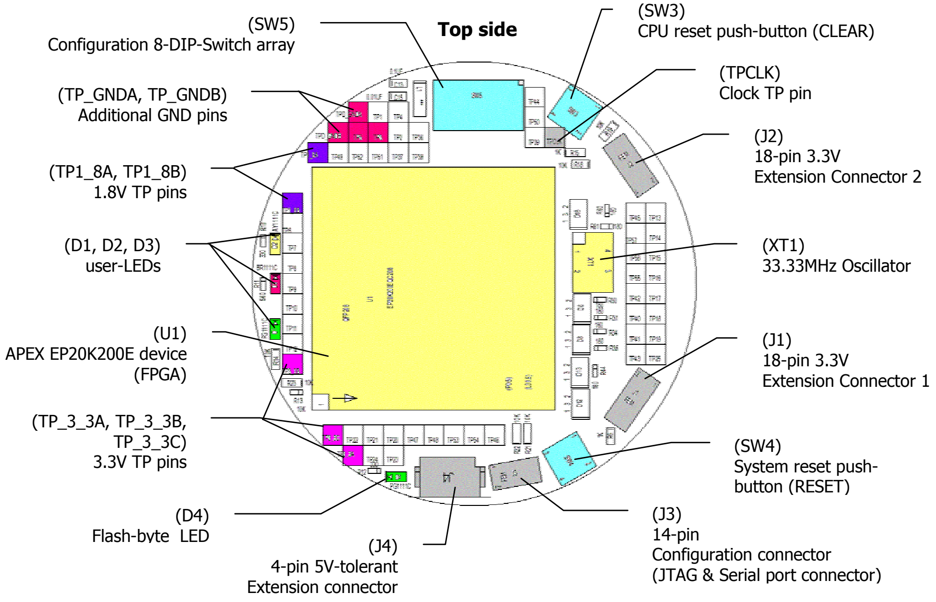


Figure 5 : TOP-side FPGA Board component layout

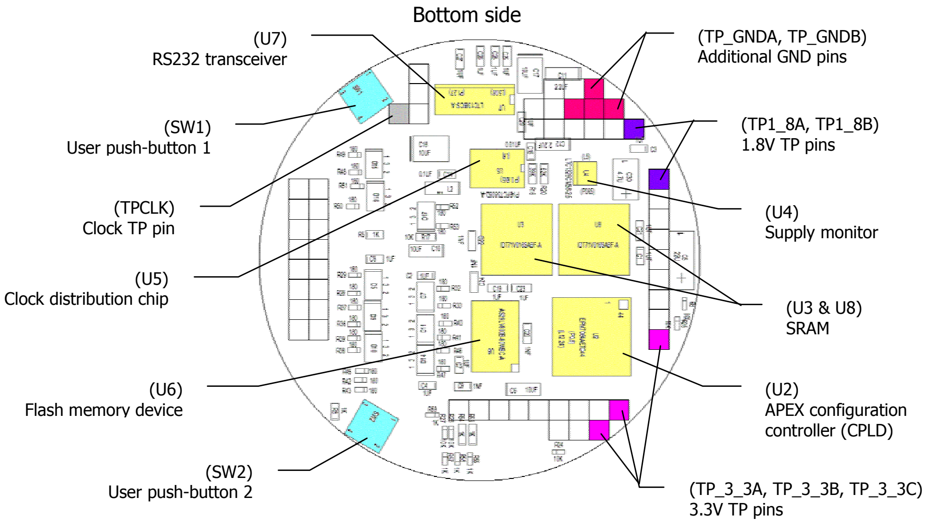


Figure 6 : BOTTOM-side FPGA Board component layout

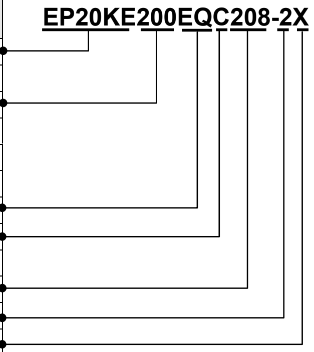
2.4 Component description

2.4.1 FPGA

U1 is an APEX 20K200E device whose characteristics are given in the next table. It can be configured with two separate methods:

- A JTAG interface can be used with Quartus II software via a programming cable
- A configuration controller (U2) that configures the device at power-up from hexout files stored in the flash memory (U6).

EP20KE200EQC208-2X characteristics	
Device Family	APEX 20KE
Maximum system Gates	526 000
Typical Gates	200 000
Maximum System Gates	525 824
Logic Elements	8 320
Maximum RAM Bits	106 496
Package	PQFP
Temparture range	0-80°C (commercial)
Maximum User I/O Pins	136
Total Pins	208
Speed grande	2
PLLs	Yes
Size [mm]	30.4 x 30.4
Cost [US$]	140 $



2.4.2 Flash memory

U6 is an advanced micro-devices (AMD) AM29LV800BB 1Mbyte flash memory chip. It is connected to the APEX device so that it can be used for two purposes simultaneously:

1. A Nios processor implemented on the APEX device can use the flash as general-purpose readable, writable non-volatile memory.

2. The flash memory can hold an APEX device configuration file that is used by the configuration controller to load the APEX device at power-up.

A hexout configuration file that implements the 32-bit Nios reference design is pre-loaded in this flash memory. The 32-bit reference design, once loaded, can identify the 1 MB flash in its address space, and includes monitor software that can download files (either new APEX device configurations, Nios software, or both) into flash memory. The Nios SDK includes subroutines for writing and erasing this specific type of AMD flash memory.

2.4.3 SRAMs

U3 and U8 (IDT71V016SA12BF) are 256 Kbytes (64K x 16-bit) asynchronous SRAM chips. They are connected to the APEX device so that they can be used by a Nios processor as general-purpose zero-wait-state memory. The SRAMs can be configured for use with either 16-bit (64K x 16) or 32-bit (64K x 32) applications.
The two 16-bit devices can be used in parallel to implement a 32-bit wide memory subsystem. The Nios 32 bit reference design identifies these SRAM chips in its address space as a contiguous 256 Kbytes, 32-bit-wide, zero-wait-state main memory. The Nios 16 bit reference design uses only one memory chip.

2.4.4 3.3V-Extension connectors

J1 and J2 are compact SAMTEC 18-pins female black connectors that can be used as an interface to a user board for example. These connectors can drive a 5V-logic device, but a 5V-logic device cannot drive the APEX device (unless a clamping diode is added with a serial resistance, see APEX 5V-tolerance White Paper).

Each 3.3V extension connector interface includes:

- 13 APEX device general-purpose I/O signals
- A buffered, zero-skew copy of the on-board oscillator (from U5)
- A buffered, zero-skew copy of the APEX PLL's output (from U5)
- A 5V power supply pin (connected physically to VCC_EXT Khepera power pin)
- A regulated 3.3 V power supply pin
- A ground connection

The user should be careful with pins numeration, because it unhappily doesn't begin on the same side for both J1 and J2 connectors!

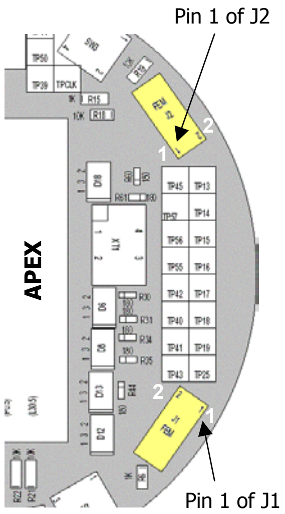


Figure 7: 3.3V extension connectors J1 and J2

J2

Signal	Pin	Pin	Signal
+3.3V	2	1	+5V
clk_APEX0	4	3	GND
(FAST) Ext<0> (181)	6	5	clk_Osc2
(FAST) Ext<2> (77)	8	7	(FAST) Ext<1> (81)
Ext<4> (116)	10	9	Ext<3> (119)
Ext<6> (88)	12	11	Ext<5> (89)
Ext<8> (85)	14	13	Ext<7> (87)
Ext<10> (124)	16	15	Ext<9> (84)
Ext<12> (113)	18	17	Ext<11> (117)

Figure 8 : Signals on J2 connector

J1

Signal	Pin	Pin	Signal
+3.3V	2	1	+5V
clk_APEX1	4	3	GND
Ext<13> (13)	6	5	clk_Osc3
Ext<15> (15)	8	7	Ext<14> (14)
Ext<17> (116)	10	9	Ext<16> (119)
Ext<19> (88)	12	11	Ext<18> (89)
Ext<21> (85)	14	13	Ext<20> (87)
Ext<23> (124)	16	15	Ext<22> (84)
Ext<25> (113)	18	17	Ext<24> (117)

Figure 9 : Signals on J1 connector

2.4.5 5V-tolerant extension connector

J4 is a 5V-tolerant little MicroMaTch 90° 4-pin connector. It has only two general 5V I/O and two GND pins. It can be used as an additional 5V serial line (with RxD and TxD lines). The serial resistances should be adapted according to the connected 5V-device. 3.3V devices can drive the APEX through this interface without any restrictions.

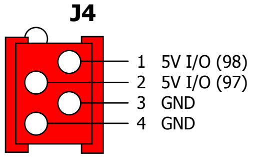


Figure 10: Signals on J4 connector

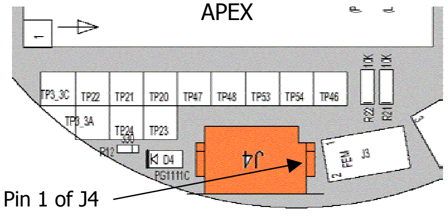


Figure 11: J4 connector position

2.4.6 Configuration/Download connector

On the Excalibur development board, the serial port connector and the JTAG connector are separated, but in the FPGA board, because of limited area, they are combined in one compact SAMTEC 14-pin female connector. The first eight pins are dedicated to the JTAG part and the other pins are dedicated for the serial port. A special interface adapter allows the user to connect both, the ByteBlasterMV download cable on the JTAG part and the serial port cable (see figure 14).

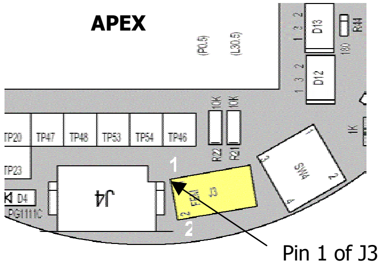
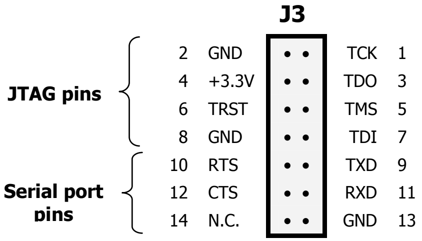


Figure 12: J3 connector position



2	GND		TCK	1
4	+3.3V		TDO	3
6	TRST		TMS	5
8	GND		TDI	7
10	RTS		TXD	9
12	CTS		RXD	11
14	N.C.		GND	13

Figure 13: Signals on J3 pins

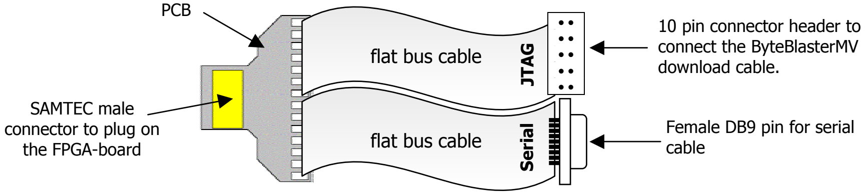
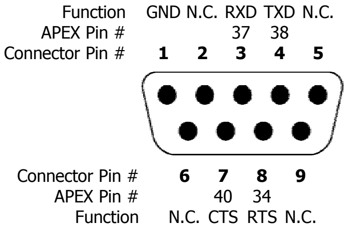

PCB
flat bus cable
JTAG
flat bus cable
Serial
SAMTEC male connector to plug on the FPGA-board
10 pin connector header to connect the ByteBlasterMV download cable.
Female DB9 pin for serial cable

Figure 14 : Interface cable

2.4.6.1 Serial Port connector part

The serial port connector part is typically used for host communication with a desktop workstation using a 9-pin serial cable connected to a COM-port. The transmit (TXD), receive (RXD), clear to send (CTS), and ready to send (RTS) signal, use RS-232 standard high-voltage levels. U7 is a level-shifting buffer that presents or accepts 3.3V versions of these signals to and from the APEX device.



Function	GND	N.C.	RXD	TXD	N.C.
APEX Pin #			37	38	
Connector Pin #	1	2	3	4	5
Connector Pin #		6	7	8	9
APEX Pin #			40	34	
Function		N.C.	CTS	RTS	N.C.

Figure 15: Signals on the serial port connector pins of the interface cable

2.4.6.2 JTAG connector part

The JTAG connector on the interface adaptor is compatible with Altera ByteBlasterMV and MasterBlaster programming cable. The JTAG connection can be used for one of two purposes:

1. Quartus II software can configure the APEX device (U1) with a new bitstream file (*.sof) via a MasterBlaster or ByteBlasterMV programming cable.
2. Quartus II or MAX+PLUS II software can re-program the EPM7064 device (U4) with a new *.pof file via a MasterBlaster or ByteBlasterMV programming cable.

For each device, two switches connect or unconnect the TDI and TDO JTAG lines. Only one device can be configured at the same time with a new JTAG chain. The four switches on the switch array determine which device is connected for configuration through its JTAG pins:

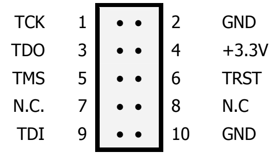
Table 1: Switch array configuration for JTAG connection on FPGA

	8-DIP-Switch array (SW5)			
Selected device	**n°1**	**n°2**	**n°3**	**n°4**
APEX (FPGA)	0	1	0	1
EPM7064 (CPLD)	1	0	1	0

The JTAG connection is most commonly used to download user configuration files (*.sof) to the APEX device chip during logic development and debugging. In this case, it is usually most convenient to leave switches n°2 and n°4 in the connect position, and switches n°1 and n°3 in the unconnected position.
The EPM7064 device (U2) is programmed as a configuration controller. Most users will never need to reprogram the configuration controller. Re-programming U2 may result in an inoperable FPGA board. Therefore, it's strongly recommended to leave switches n°1 and n°3 in the unconnected position!

The design, implementation and programming files for the configuration-controller are available at user's disposal.

JTAG connector on the interface adapter:



TCK	1		2	GND
TDO	3		4	+3.3V
TMS	5		6	TRST
N.C.	7		8	N.C
TDI	9		10	GND

Figure 16: Signals on the JTAG connector of the interface cable

2.4.7 Configuration controller

The configuration controller (U4) is an Altera Max7000 device family PLD (EPM7064AE). It was factory-programmed with logic that configures the APEX device (U1) from data stored in flash (U3) on power-up. On power-up (or when the reset switch SW2 is pressed), the configuration controller begins reading data out of the flash memory. The flash memory, APEX device, and configuration controller are connected so that data from the flash configures the APEX device in passive-parallel mode.

2.4.7.1 Configuration Data

The Quartus II software can (optionally) produce hexout configuration files, which are directly suitable for download and storage in the flash memory as configuration data. A hexout configuration file for the APEX20K200E device (U1) is a little less than 256Kbytes, and thus occupies about 1/4 of the flash memory (U3).

New hexout can be stored in the flash memory (U3) by software running on a Nios processor. The preloaded 32-bit Nios reference design includes the GERMS monitor program, that supports downloading hexout files from a host (e.g., desktop workstation) into flash memory. See the Nios Embedded Processor Software Development User Guide for a detailed description of the GERMS monitor program.

2.4.7.2 Factory and User Configurations

The configuration controller can manage two separate APEX device configurations stored in flash memory. These two configurations (hexoutfiles) are conventionally referred to as the user configuration and the factory configuration. Upon reset (or when the reset switch (SW2) is pressed), the configuration controller will attempt to load the APEX device with user configuration data. If this process fails (either because the user-configuration is invalid or not present) the configuration controller will then load the APEX device with factory configuration data.

The configuration controller expects user-configuration and factory-configuration files to be stored at fixed locations (offsets) in flash memory. The following table (Table 2) shows how the configuration controller expects flash memory contents to be arranged:

Table 2:Flash Memory Allocation

Flash Memory Allocation		
0x100000 – 0x17FFFF	512 Kbytes	Nios instruction and nonvolatile data space.
0x180000 – 0x1BFFFF	256 Kbytes	User-defined APEX device configuration data.
0x1C0000 – 0x1FFFFF	256 Kbytes	Kbytes Factory-default APEX device configuration

The 32-bit Nios reference design is pre-loaded into the factory-configuration region of the flash memory. It is recommends that users avoid overwriting the factory configuration data.

The switch n°5 of the 8-DIP-Switch array (SW5) changes the behavior of the configuration controller. If the switch is "ON", the configuration controller will ignore the user-configuration and always configure the APEX device from the factory configuration. The switch allows the user to "escape" from the situation where a valid-but-non-functional user configuration is present in flash memory.

In the pre-loaded Nios reference design, the 1 MB flash memory is mapped at base-address 0x100000. Thus, user hexout-files should be downloaded to address 0x180000 (= flash-base-address + user-configuration offset).

2.4.8 8-DIP-Switch array

This switch array is used to configure different options on the FPGA board. The function of each switch is given in the table 3

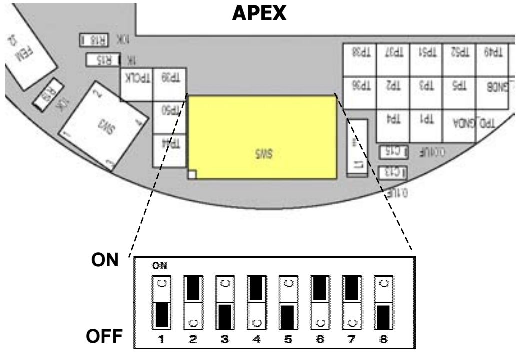


Figure 17: 8-DIP-Switch array position

Table 3 :

Function name	Description	Switch n°
JTAG target device selection	If "ON", the JTAG connector pin "TDO" is connected to the MAX7064 (CPLD)	1
	If "ON", the JTAG connector pin "TDO" is connected to the APEX (FPGA)	2
	If "ON", the JTAG connector pin "TDI" is connected to the MAX7064 (CPLD)	3
	If "ON", the JTAG connector pin "TDI" is connected to the APEX (FPGA)	4
Configuration file selection	If "ON", the default FPGA configuration on the flash is charged on the FPGA at next power-up sequence, even if there is a user configuration in the second part of the flash.	5
Auto-configuration ability	If "ON", the auto-configuration ability of the Apex is activated.	6
Clock circuitry configuration	If "ON", connect the oscillator clock signal to the on-board devices	7
	If "ON", the click output or input is enabled.	8

2.4.9 Push-buttons

SW1 (91), SW2 (90) and SW4 (184) are momentary-contact push-button switches. Each is connected to an APEX device general-purpose I/O and a pull-up resistor. The APEX device will see a logic-0 when each switch is pressed.

SW1 and SW2, which can be found on the reverse side of the FPGA-board, are user-definable. The other switches (SW3 and SW4) are dedicated and have the following fixed functions:

2.4.9.1 SW3: CLEAR

When SW3 is pressed, a logic-0 is driven onto the APEX devices' DEV_CLRn pin (and user I/O 184). The result of pressing SW3 depends on how the APEX device is currently configured. The pre-loaded Nios reference design treats SW3 as a CPU-reset pin: The reference Nios CPU will reset and start executing code from its boot-address (0) when SW2 is pressed.

2.4.9.2 SW4: RESET

When SW4 is pressed, a logic-0 value is driven to U7, the power-on reset controller. Pressing SW4 is equivalent to a power-on reset. When SW4 is pressed (or when the board is power-cycled), the configuration controller will load the APEX device from flash memory. See §2.4.7 more information. When the development board is delivered from the factory, the APEX device will be configured with the 32-bit reference design at power-up (or when SW4 is pressed). The reference design will then begin executing the GERMS monitor, a serial debug/download utility.

2.4.10 LEDs

2.4.10.1 User LEDs

Three users LEDs (D1, D2, D3) of different colours are each controlled by a general-purpose I/O of the Apex device. Each LED will light up when the APEX device drives a logic-1 on its controlling output.

Table 4: LEDs color

LED name	Coloured	APEX pin
D1	Green	33
D2	Yellow	32
D3	Red	30

2.4.10.2 Dedicated LED

The green LED (D4) indicates that the configuration controller has finished configuring the APEX EP20K200E device.

This LED does not indicate successful configuration of the APEX device, but only that the configuration controller has finished sending data to the APEX device.

2.4.11 Clock circuitry

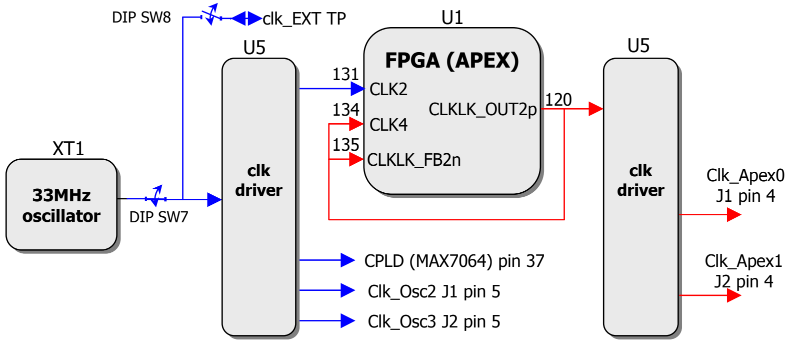


Figure 18 : Clock circuitry

The FPGA board includes a 33.33MHz oscillator and a zero-skew clock distribution networks.

- The first network (drawn in blue) is generated by the board's own oscillator or an external one. Components are driven by the clock distribution chip (U5). The oscillator is driven to the APEX (pin 131), the configuration controller (U2), and the two extension connectors (J1 and J2).

- The second type of clock network (drawn in red) carries a signal produced by the phase-locked loop circuitry on the APEX EP20K200E device. The user has the option of producing a clock with the PLL circuitry by the use of the *altcklock* megafunction in the Quartus II software (see ClockLock & ClockBoost PLL Features in §2.4.13), which may be driven off-chip via pin 120 (CLKLK_OUT2p). The signal is also fed to the clock distribution chip (U5), and fed to both main extension connectors (J1 & J2).

The oscillator may be replaced at the user's discretion, but the configuration controller design may fail to successfully configure the APEX EP20K200E device if the clock frequency is greater than 66.8MHz.
If the user oscillator's frequency is not 33.33MHz (but less than 66.8MHz), the CPLD configuration should be adapted according to comments in "dclk_divider.tdf" source file.

In a single FPGA board use, switches n°7 (of SW5 switch array) and should be on and n°8 off. These switches are needed to configure the clock circuitry architecture in multiple FPGA-boards use (§2.4.14).

The FPGA has four clock inputs (CLK1 to CLK4), but only two inputs are used in the actual design.

2.4.12 Apex Self-configuration optional feature

Sophisticated Apex designs (e. g. CPU systems) might wish to reconfigure themselves. That functionality is provided by the Apex_reload_n input (pin35) in the MAX7064 (CPLD). If this pin is driven low (0V), the configure-from-flash sequence will restart.

A pull-up resistance puts the Apex_reload_n signal in logic high level by default. If an APEX user implementation always drives the Apex_reload_n low by error, the system eternally reconfigures itself. A dedicated switch (switch n°6 on SW5) (cf. figure 19) allows the user to inactivate (switch in off-position) the self-configuration ability in such a situation in order to stop the process and then download a new configuration.

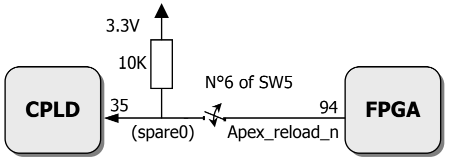


Figure 19 : Self-configuration circuitry

2.4.13 ClockLock & ClockBoost PLL features

Some APEX devices (with a “X” suffix after the speed grade), like the EP20K200EQC208-2X on the FPGA-board, have *ClockLock* & *ClockBoost* PLL features.

The *ClockBoost* feature is used in conjunction with the *ClockLock* phase-locked loop feature. It can be used to generate internal clocks that operate at frequencies that are multiples of the frequency of the system clock. The *ClockBoost* feature also provides clock delay reduction.
Programmable clock delay and phase shift is also provided by a *ClockShift* circuitry with a resolution range of 0.4 ns to 1.0 ns.

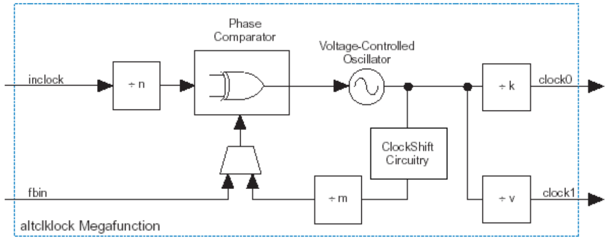


Figure 20 : Schematic view of the altcklock megafunction

ClockLock and *ClockBoost* features are available in Quartus with the help of the altclklock megafunction.

In the actual APEX device, up to two PLLs can be implemented (see figure 20). Each PLL includes circuitry that provides clock synthesis for two outputs using m/(n x k) and m/(n x v) scaling factors (m, n and v are integer values) When a PLL is locked, the locked output aligns to the rising edge of the input clock.

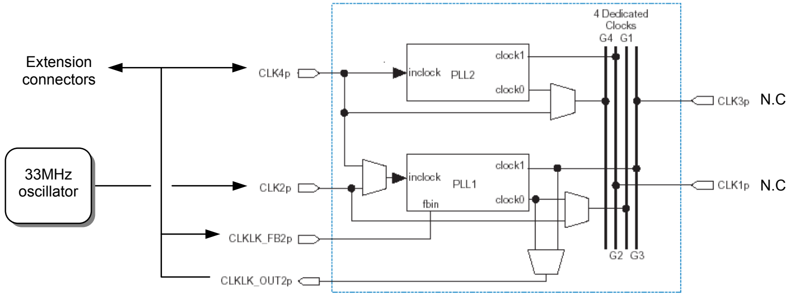


Figure 21 : FPGA’s PLLs architecture

A generated clock might be driven off-chip via pin 120 (CLKLK_OUT2p) which are fed to the main extension connectors (J1 & J2).

For a complete and detailed description of the features associated with altclklock megafunction, see the Altera application note (AN 115).

2.4.14 Multiple FPGA Boards configuration

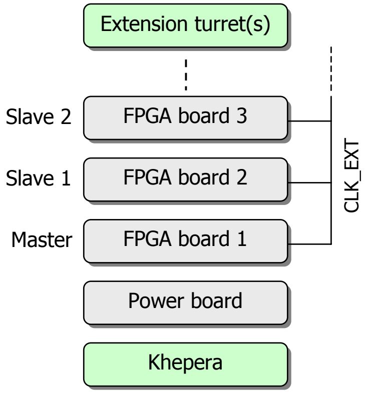


Figure 22 : Multiple FPGA Boards configuration

To improve the resources, the FPGA module can be used with two or more FPGA boards in master/slave architecture as shown in figure 22. In such architecture, each FPGA requires a common clock signal in order to synchronize the communication between the devices. In this configuration, the master oscillator's output should be tied to all FPGAs clock's input. The data, indeed, has to be latched or read on the same clock rising edge.

A dedicated supplementary TP pin called CLK_EXT can be used either as a clock input, either as a clock output or as a non-connected pin. As all FPGA boards have the same physical connection with their CLK_EXT pin, it is important to configure correctly the pin property on each board.

The switches n°7 and n°8 of SW5 allow the user to configure the clock circuitry mode:

- **Single FPGA board mode.** (cf. figure 23) The oscillator drives the devices on the board through the clock driver. The CLK_EXT pin is unconnected. Switches configuration:
  - **switch n°7: on**
  - **switch n°8: off**

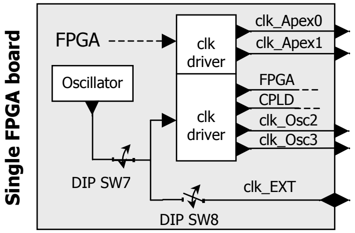


Figure 23 : Clock signal connections in Single FPGA board configuration

- **Multiple FPGA boards mode: MASTER.** The oscillator drives the devices on the board through the clock driver and the CLK_EXT (clk output)**.** Switches configuration:
  - **switch n°7: on**
  - **switch n°8: on**

- **Multiple FPGA boards mode: SLAVE.** The oscillator is unconnected and the devices on the board are driven by the CLK_EXT pin (clk input) through the clock driver. Switches configuration:
  - **switch n°7: off**
  - **switch n°8: on**

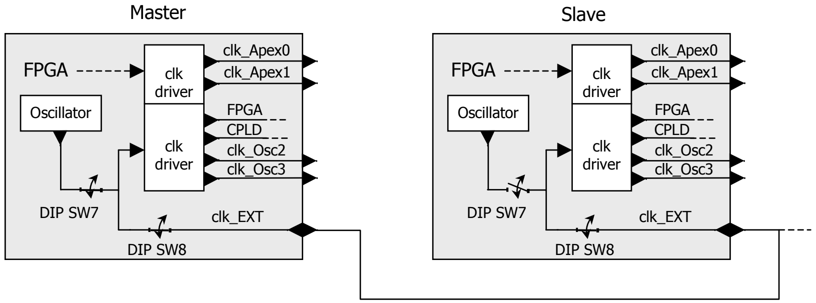


Figure 24 : Clock circuit in a Multiple FPGA Boards in a master/slave configuration

If both switches are off, no clock signal feeds the board's devices and the FPGA module cannot operate. It is recommended to avoid driving devices with more than one oscillator. This may alter the module's functionality. The communication between the modules can be performed with the Khepera bus pins or through the extension connectors.

As many slave FPGA boards can be used, unless the power board is unable to power all FPGA boards. Multiple FPGA-boards feature gives a high level of flexibility.

Table 5: Resume of the modes and their configuration on 8-DIP Switch array (SW5)

Mode	N°7	N°8
Single FPGA board mode	1	0
Multiple FPGA board mode: MASTER	1	1
Multiple FPGA board mode: SLAVE	0	1

3. Power Board Turret

3.1 Power Board architecture

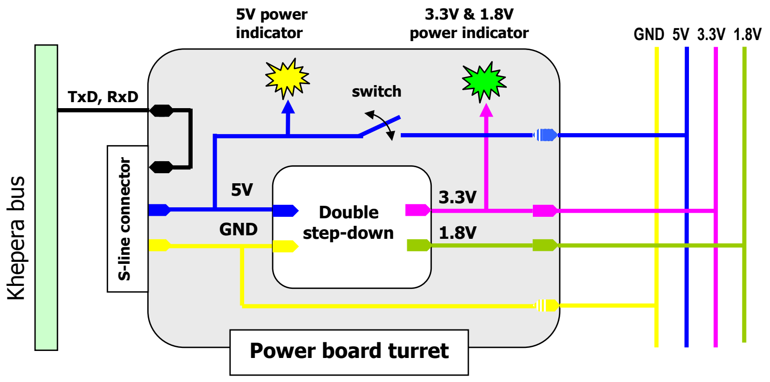


Figure 25 : Power Board architecture

3.2 Features

- 1.8V and 3.3V general use TP connectors with maximum load of 1.4A for each voltage level.
- Power supply: 4.5V to 25V (5V-Khepera compatible)
- Power indicator LEDs: green (1.8V and 3.3V) and yellow (5V)
- Khepera-compatible 6-pins connectors for separate external power input
- A switch to connect the separate external input power to the Khepera 5V TP connector (VCC_Ext)
- Ability to connect any extension turret above and in particular up to three FPGA boards (under some conditions).

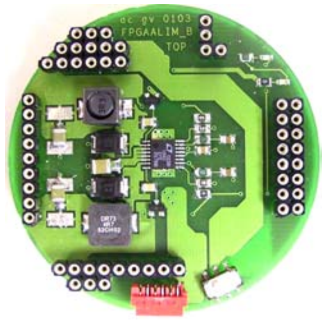


Figure 26 : Power Board picture (scale 1:1)

3.3 Layout & Components

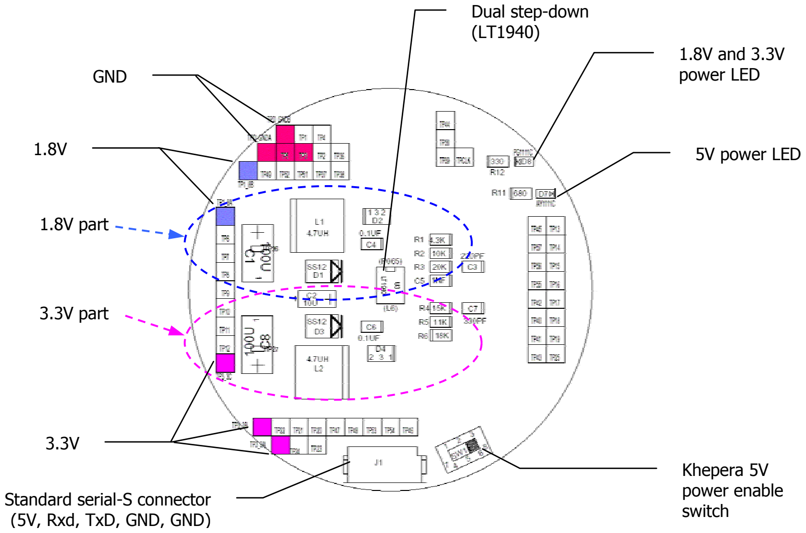


Figure 27 : Power Board layout of components

At first sight, the power PCB seems symmetrical (cf. figure 27). The dual step-down is placed in the center and on each side are the parts related with the correspondent step-down (inductance, output capacitor, resistor divider, etc.)

3.3.1 1.8V, 3.3V and GND outputs

The outputs of generated voltages are connected to supplementary TP pins. The 1.8V is available on two pins and the 3.3V on three pins. Multiple pins for the same tension prevents the voltage from decreasing due to the serial resistance contact between the female TP pin on that board and the male TP pin of the plugged FPGA turret (or any other turret).
For the same reason two GND TP pins are added. In the whole, there are four GND TP pins.

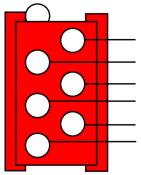
3.3.2 Power indicator LEDs

On the edge of the board, two LEDs indicate to the user the presence of voltages:

- A yellow LED (D7) informs about the presence the input voltage (5V)
- A green LED (D8) is connected on the 3.3V output and therefore indicates the presence of a voltage on the 3.3V output, and indirectly on the 1.8V too (because of the output sequence)

3.3.3 Standard serial-S connector

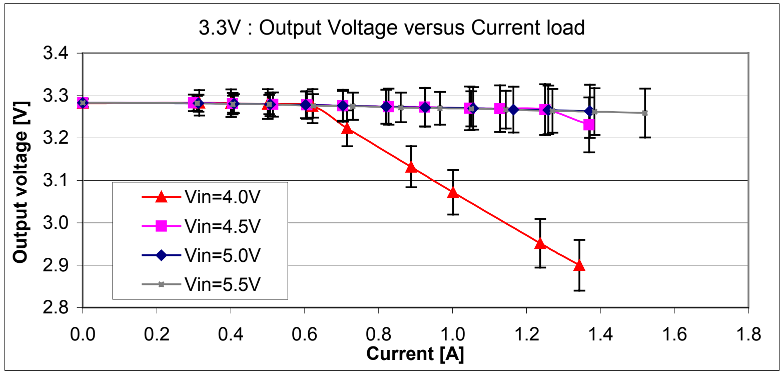
The connector is a MicroMaTch 90° with 6pins. The pins are compatible with the serial line S connector furnished with the Khepera kit:



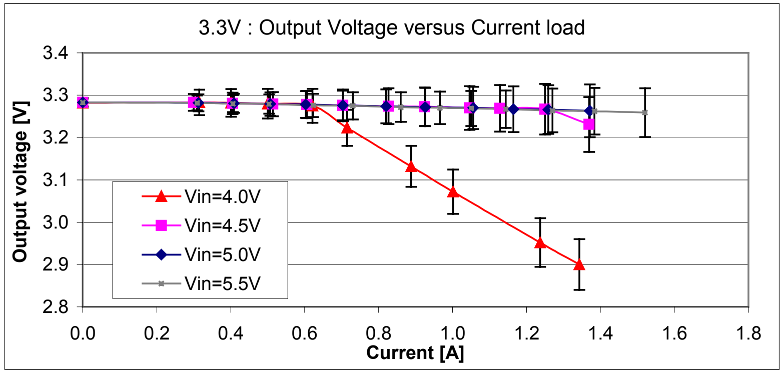
1	VCC	Power supply +5V
2	RxD	Serial receive data (TTL levels)
3	TxD	Serial transmit data (TTL levels)
4	GND	Power supply ground
5	GND	Power supply ground
6	GND	Power supply ground

Figure 27 : Signals on S-line connector

3.4 Characteristics



Graph 1 : 3.3V output voltage versus current load



Graph 2 : 3.3V output voltage versus current load

4. Demo Module Board

A demonstration Module board was realised to extend the FPGA module's user interfaces. This module can be connected on J2 3.3V extension connector.

4.1 Layout

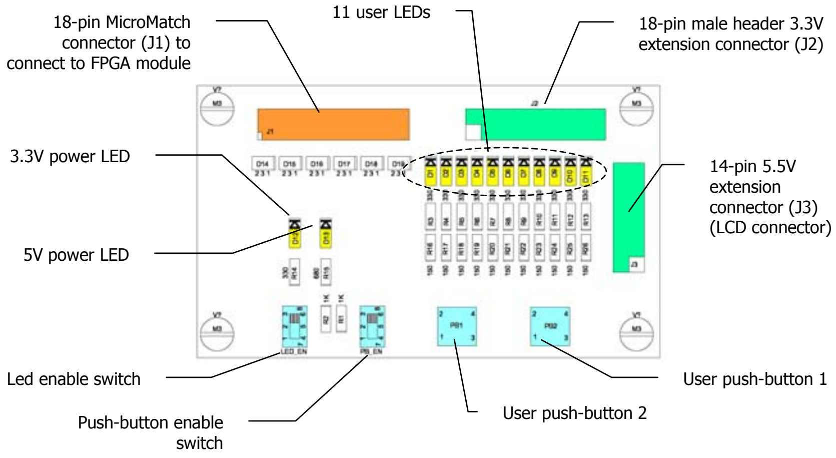

18-pin MicroMatch connector (J1) to connect to FPGA module
11 user LEDs
18-pin male header 3.3V extension connector (J2)
3.3V power LED
14-pin 5.5V extension connector (J3) (LCD connector)
5V power LED
Led enable switch
User push-button 1
Push-button enable switch
User push-button 2

Figure 28 : Demo Module layout

4.2 Features

- 11 user LEDs (D1 to D11)
- Power indicating LEDs (3.3V:D12 and 5V: D13)
- 2 user push-buttons (PB1 & PB2)
- A MicroMatch 18-pin female connector (J1) to connect the J3 connector of FPGA module (with an interface cable)
- A 18-DIP extension male header (J2) witch same pins order than extension connector J3 of FPGA_module
- A 14-DIP 5V-tolerant male connector (J3) compatible with the Excalibur Board's LCD module.
- A LED enable switch (LED_EN)
- A push-button enable switch (PB_EN)

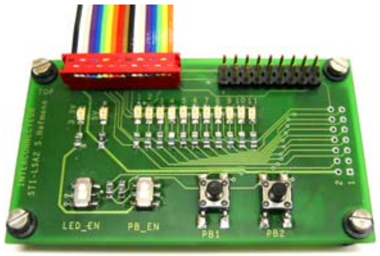


Figure 29 : Picture of Demo Board

4.3 Demo Board components

18-pin female MicroMatch connector (J1)

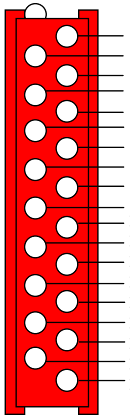
This connector must be connected with a specific interface connector to only J3 3.3V extension connector of FPGA module for pin compatibility.

Push-buttons & PB enable switch

As in the FPGA module, push-button switches are connected to a pull-up resistor. A switch (PB_en) disconnects these resistors, which allows using pin 17 and 18 as a normal user I/O on connector J2.

LEDs

The 11 user LEDs can be disconnected by a switch too (LED_en). Nevertheless, signals Ext<10..0> on J2 connector can be used in any case (LEDs on / LEDs off).



1	5V	
2	3.3V	
3	GND	
4	clk_Apex	
5	clk_Osc	
6	Ext<0>	LED1
7	Ext<1>	LED2
8	Ext<2>	LED3
9	Ext<3>	LED4
10	Ext<4>	LED5
11	Ext<5>	LED6
12	Ext<6>	LED7
13	Ext<7>	LED8
14	Ext<8>	LED9
15	Ext<9>	LED10
16	Ext<10>	LED11
17	Ext<11>	PB1
18	Ext<11>	PB2

Figure 30: Signals on J1 connector pins

Interface connector cable

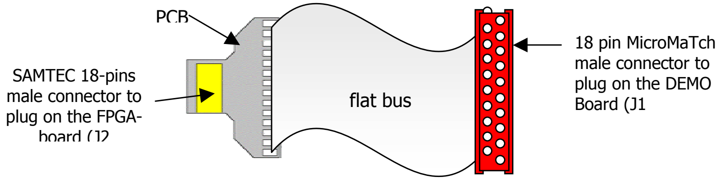

PCB
SAMTEC 18-pins male connector to plug on the FPGA-board (J2
flat bus
18 pin MicroMaTch male connector to plug on the DEMO Board (J1

Figure 31 : Interface connector cable

5V-header / LCD connector (J3)

The power pins of these connector are compliant witch the LCD module if the Excalibur Development Board.

All signal pins on this connector have a serial resistance of 150Ω and clamping diode are connected to 3.3V as on the FPGA module.

J3

GND	1	2	+5V
N.C.	3	4	Ext<0>
Ext<1>	5	6	Ext<2>
Ext<3>	7	8	Ext<4>
Ext<5>	9	10	Ext<6>
Ext<7>	11	12	Ext<8>
Ext<9>	13	14	Ext<10>

Figure 32: Signals on J3 connector pins

3.3V extension connector (J2)

J2

+5V	1	2	+3.3V
GND	3	4	clk_Apex
Clk_Osc	5	6	Ext<0>
Ext<1>	7	8	Ext<2>
Ext<3>	9	10	Ext<4>
Ext<5>	11	12	Ext<6>
Ext<7>	13	14	Ext<8>
Ext<9>	15	16	Ext<10>
Ext<11>	17	18	Ext<12>

Figure 33: Signals on J2 connector pins

5. Typical FPGA Development Flow

This chapter is user-oriented and gives typical stages for new design implementation on the FPGA board turret with or without the Khepera.

5.1 Reference design selection

Four different reference designs are available with a Nios Processor (see table 6). These designs can be found in the *FPGA_module_ref_designs* directory.

Table 6 : Reference designs and their features

Reference design	all_on_chip_16	all_on_chip_32	reference _design_16_bit	reference _design_32_bit
Processor	16-bit Nios	32-bit Nios	16-bit Nios	32-bit Nios
Serial ports	UART1: 115'200 baud (used to download executable files with PC) UART2: 38'400 baud (connected on Khepera's serial port)			
ROM	Internal ROM with resident "GERM" boot-monitor program			
RAM	Internal RAM (4Kbytes)		Interface to 32KByte of 0-wait-state external SRAM	Interface to 256KByte of 0-wait-state external SRAM
Flash			Interface to 1Mbyte external Flash memory	
Parallel I/O ports			LEDs, buttons, extension connectors, etc.	

The user can start his design from one of these reference designs. All user I/O signals are grouped in a schematic file (FPGA_module_pins.bdf) and can be copied and added in the design.

5.2 Communication protocols with Khepera

There are three ways to communicate with the Khepera:

- **Serial Line**. It's a simple way to communicate with the FPGA in a Master mode. In the actual reference designs, only that communication mode is used. A distinct UART port is connected with the Khepera's one.
- **K-NET bus.** This is a SPI based bus but it allows to addressing many different turrets like the I^{2}C bus. (See K-NET bus documentation from K-team). In such a configuration, all extension turrets are in Slave mode.
- **Address decoding.** The FPGA module works directly with the Khepera addresses and data.

The communication controller on the FPGA module should be implemented with program, which is executed by the Nios. Hardware implementation on the FPGA is also possible but requires more time.

5.3 Design download on FPGA

After the design has been successfully compiled, it can be downloaded with JTAG interface in the FPGA for testing and evaluation. (Quartus provides also a simulation tool).
Executables files (*.srec) can also be downloaded with the serial line connector with help of Bash terminal (nios-run command), but C/C++ file should be previously compiled with the nios-build command.

Debugging can be performed either by sending data on the bash terminal or by using the SignalTap functionality in Quartus. SignalTap allows capturing functional data (with JTAG) while a device is operating.

5.4 Design download in flash

Once successfully tested, the users can download the design on the flash so the FPGA configures automatically with the user design on power-up or on system reset.

To perform this stage, the reference_design_32_bit should be first be implemented on the FPGA with a GERMS monitor which allows the user to download a configuration file in the flash. But before, the configuration file should be converted in an adequate format (*.flash): *hexout2flash* routine converts the *.hexout configuration file in the adequate format. The file can then be downloaded in flash with the *nios-run* command (automatically at 0x180000 flash memory location) and the design becomes the default booting design for the FPGA module.

Executable file can also be stored on the flash. The GERMS monitor will automatically execute code from flash to SRAM after initialisation. (see srec2flash routine)
A detailed description is given in the Nios Embedded Processor Software Development Reference.